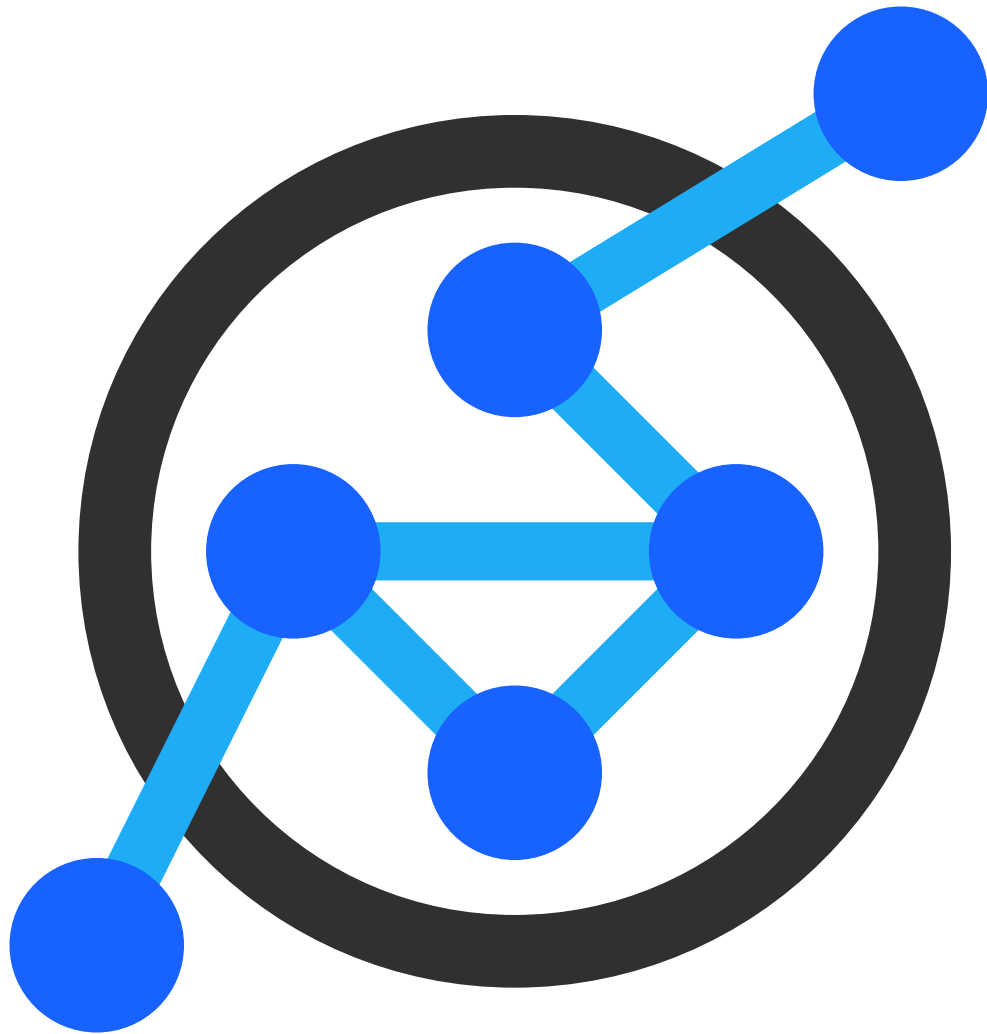


Zweite Runde Theorie

Aufgaben



Swiss Olympiad in Informatics

4. – 7. März 2022

Anweisungen

- Du hast 5 Stunden Zeit, um die Aufgaben zu lösen. Stelle sicher, dass du genügend Zeit zum Einsenden einplanst. Wir empfehlen dir, dass du 30 Minuten vor Schluss beginnst, deine Lösungen einzuscannen und hochzuladen. Wenn du danach noch Zeit hast, kannst du weiterarbeiten und eine verbesserte Version später noch hochladen.
- Der Wettbewerb ist "Open Book"; du darfst das Internet, Bücher, alte Lösungen, und so benutzen.
- Falls du Fragen während der Runde hast, frage uns via E-Mail (info@soi.ch). Wir geben dir keine Hinweise, wie die Aufgabe zu lösen ist, deshalb versuche, klare Ja/Nein-Fragen zu stellen.
- Deine Lösung muss vollständig von Hand geschrieben werden (mit Papier und Stiften).
- Schreibe lesbar.
- Deine Lösung sollte keine externe Quellen (bekannte Algorithmen, Blog-Artikel, Papers) referenzieren, sondern alles von Grund auf erklären. Ausgenommen davon sind das, was im SOI Wiki in den Abschnitten "First Round" und "Second Round" steht oder vorausgesetzt wird.
- Du sollst ein PDF pro Aufgabe hochladen. Stelle sicher, dass die Auflösung gut genug ist, dass wir alles lesen können. Einige Megabytes pro Aufgabe sollte ausreichen (wir setzen eine Grenze von 100 MiB).
- Wir schauen uns nur die letzte Einsendung jeder Aufgabe an. D.h. du kannst anfangs eine Version hochladen und diese später aufbessern.
- Du kannst eine Vorschau deiner Einsendung sehen, wenn du auf "Detail" klickst. Das erlaubt dir sicherzustellen, dass du die richtige Datei hochgeladen hast.
- Solltest du technische Probleme mit deinem Scanner oder dem Grader haben, kannst du uns notfalls die Aufgaben auch manuell und in geringerer Auflösung schicken. Sende diese an Joël Huber.

Aufgaben

Beachte die Reihenfolge der Aufgaben: Die erste Aufgabe ist als die einfachste Aufgabe gedacht. Die anderen drei Aufgaben sind in einer zufälligen Reihenfolge angeordnet. Darum empfehlen wir dir, alle Aufgaben durchzulesen und über alle Aufgaben nachzudenken.

Für jede Aufgabe kannst du annehmen, sofern es in der Beschreibung nicht anders angegeben ist, dass die Eingabe in Form von *schreibgeschützten* Variablen und Arrays, welche *nicht* zur Speichernutzung deiner Lösung beitragen.

Bewertung

Deine Lösung wird anhand ihrer Korrektheit und ihrer asymptotischen Laufzeit und Speichernutzung bewertet, sowie der Begründung dieser Punkte. Wir erwarten, dass du einen Beweis oder eine Beweisskizze für die Korrektheit und Laufzeit/Speichernutzung lieferst.

Du kannst jederzeit auf den Inhalt des SOI Wikis und 2H verweisen. Du musst nicht erklären, wie z.B. Dijkstra funktioniert, aber solltest begründen, wo und wieso man ihn anwenden kann. Im Fall von Dijkstra muss der Graph klar definiert sein und seine Kantengewichte dürfen nicht negativ sein.

Der Algorithmus sollte genug detailliert beschrieben sein, dass man ihn anhand der Erklärung programmieren könnte. Wichtig ist hier auch, dass alle verwendeten Datenstrukturen genau beschrieben werden. Am einfachsten ist es erfahrungsgemäss, kurz einen Pseudocode hinzuschreiben.

Wir empfehlen dir, dich an folgende Struktur zu halten:

1. Beschreibe die Idee hinter deinem Algorithmus so verständlich wie möglich.
2. Gib Pseudocode oder erkläre, wie man den Algorithmus implementieren könnte.
3. Begründe, wieso der Ansatz die Aufgabe korrekt löst.
4. Analysiere und begründe die asymptotische Laufzeit und den asymptotischen Speicherverbrauch.

Wenn ein Teil deiner Lösung für mehrere Teilaufgaben gilt, reicht es, ihn nur einmal aufzuschreiben und von da an auf ihn zu verweisen. Beachte aber, dass ein allgemeiner Algorithmus die vorherigen Teilaufgaben zwar lösen kann, aber nicht zwingend optimal ist.

Konsens

Für Maus Stofl ist das Sbrinz-Camp das Highlight des Jahres. 8 Tage lang nichts ausser Algorithmen, programmieren und extra harter Käse. Fantastisch!

Nur leider ist für Stofl das Camp viel zu kurz. So ist er zu jedem der n Leiter gegangen und hat sie gefragt, was ihre optimale Anzahl Tage für das Camp wäre. Wie erwartet hat Stofl viele verschiedene Meinungen erhalten. Sie sagten Stofl, wenn er einen Konsens unter den Leitern erreicht, was bedeutet dass sich alle auf die gleiche Anzahl Tage einigen, dann würden sie das Camp auf diese Dauer ändern. "Herausforderung angenommen", dachte sich Stofl.

Gegeben sind n Zahlen a_0, a_1, \dots, a_{n-1} gegeben, wobei a_i die bevorzugte Anzahl Tage des i -ten Leiters ist.

Stofl kann zwischen genau k Leitern (die Konstante k ist in der Eingabe gegeben), eine *Mediation* durchführen. Dazu lädt er sie zu einer Diskussionsrunde ein, und tut dann folgendes:

- Abstimmen, ob das Camp $\geq x$ Tage dauern sollte.
Mindestens die Hälfte der anwesenden Leiter (d.h. $\geq k/2$) müssen einverstanden sein.
- Abstimmen, ob das Camp $\leq x$ Tage dauern sollte.
Mindestens die Hälfte der anwesenden Leiter (d.h. $\geq k/2$) müssen einverstanden sein.
- Wenn beide Abstimmungen durchkommen, dann ändert jeder anwesende Leiter seine Meinung dahingehend, dass x die optimale Anzahl Tage für das Camp sei.

Stofl kann so viele Mediationen durchführen, wie er möchte, solange er nach endlicher Zeit fertig ist.

Was ist die maximale Anzahl Camp-Tage, für die Stofl einen Konsens erreichen kann?

Beispiel Eingabe: $k = 3, a = [3, 1, 4, 1, 5]$.

- In der ersten Mediation könnte Stofl Leiter 0, 1 und 3 einladen und eine Einigung auf $x = 1$ erzielen. Die Meinungen wären dann $a = [1, 1, 4, 1, 5]$.
- In der zweiten Mediation könnte Stofl Leiter 0, 2 und 4 einladen und eine Einigung auf $x = 4$ erzielen. Die Meinungen wären dann $a = [4, 1, 4, 1, 4]$.
- In der dritten Mediation könnte Stofl Leiter 0, 1 und 2 einladen und eine Einigung auf $x = 4$ erzielen. Die Meinungen wären dann $a = [4, 4, 4, 1, 4]$.
- In der vierten Mediation könnte Stofl Leiter 0, 1 und 3 einladen (wie bereits in der ersten Runde) und eine Einigung auf $x = 4$ erzielen. Die Meinungen wären dann $a = [4, 4, 4, 4, 4]$.

Dadurch würde der Konsens 4 erreicht. Für diese spezifische Eingabe ist das auch die grösstmögliche Zahl.

Beispiel Eingabe: $k = 2, a = [3, 1]$. Stofl kann einen Konsens für sowohl 1, 2 als auch 3 erreichen.

Teilaufgabe 1: Löse das Beispiel (10 Punkte)

Was ist der grösstmögliche Konsens, den Stofl mit der folgenden Eingabe erreichen kann?
 $a = [1, 6, 1, 8, 0, 3, 3]$ Löse es für a) $k = 3$ und b) $k = 2$.

Schreibe für jede Mediation auf, welche Leiter du einlädst, auf welche Zahl sie sich einigen, und wie die aktuellen Meinungen danach aussehen. Du musst nicht begründen, warum deine Zahl optimal ist.



Teilaufgabe 2: Entwirf einen Algorithmus (90 Punkte)

Hilf Stofl und finde einen Algorithmus, welcher das Problem für beliebige $2 \leq k \leq n$ und $n \geq 2$ löst. Dein Algorithmus muss nur die maximale Anzahl Tage ausgeben (und nicht die verschiedenen Mediationen).

Beweise, warum dein Algorithmus korrekt ist.

Gib die Laufzeit und Speicher abhängig von n und k an. Bemerke, dass $k \leq n$, und deshalb ein Algorithmus mit einer Laufzeit von $O(n^7k)$ besser bewertet wird als einer in $O(n^8)$; hingegen ein Algorithmus mit einer Laufzeit von $O(nk^2)$ wird als langsamer bewertet als einer in $O(n^2)$ (z.B. falls $n = k$ gilt).

Du kannst Teilpunkte erhalten, wenn du folgende Teilaufgaben löst:

- a) $k = 3$ – Mediation findet zwischen genau 3 Personen statt (30 Punkte).
- b) $a_i = 8$ oder $a_i = 9$ für alle i – es gibt höchstens zwei verschiedene Meinungen (30 Punkte).

Wenn du das volle Problem optimal löst, kannst du diese Teilaufgaben natürlich überspringen; allerdings kannst du möglicherweise mehr Punkte holen, wenn du für die Teilprobleme eine schnellere Lösung findest.

Wissenschaftliche Konferenz

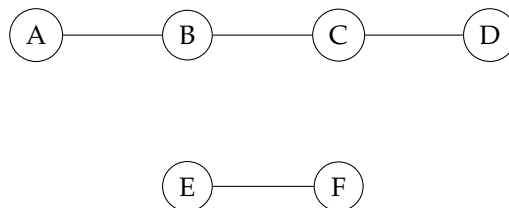
Maus Binna organisiert eine wissenschaftliche Konferenz. Es gibt n Maus-Wissenschaftler, welche sie einladen könnte. Sie weiss auch, dass m Paare von Wissenschaftlern zusammen einen Artikel veröffentlicht haben.

Für jeden Maus-Wissenschaftler, der an die Konferenz eingeladen wird, ist seine *Popularität* an der Konferenz definiert als die Anzahl anderer eingeladenen Maus-Wissenschaftler, mit denen er gemeinsam einen Artikel veröffentlicht hat.

Maus Binna weiss, dass wenn es einen eingeladenen Wissenschaftler gibt, sodass seine Popularität plus eins (wir addieren eins, um den Wissenschaftler selbst mitzuzählen) mindestens halb so gross wie die Gesamtzahl von eingeladenen Wissenschaftlern ist, dann wird sich das Geplauder an der Konferenz zu stark auf die Arbeit dieses Wissenschaftlers fokussieren. Dies wäre schädlich für das Formen neuer Ideen, Maus Binna darf dies also auf keinen Fall zulassen.

Formal ausgedrückt, wenn Maus Binna k Maus-Wissenschaftler a_1, a_2, \dots, a_k zur Konferenz einlädt, dann muss für jedes i die Ungleichung $1 + \#\{j : 1 \leq j \leq k, j \neq i, \{a_i, a_j\} \in E\} < \frac{k}{2}$ gelten, wobei E die Menge der Paare von Wissenschaftlern ist, die gemeinsam einen Artikel veröffentlicht haben. Beachte, dass diese Paare ungeordnet sind, d.h. $\{a_i, a_j\} \in E$ bedeutet dasselbe wie $\{a_j, a_i\} \in E$.

Betrachte das folgende Beispiel, wobei die Kreise Wissenschaftler repräsentieren, und Wissenschaftler, die gemeinsam einen Artikel veröffentlicht haben, durch Linien verbunden sind:



Alle diese Wissenschaftler an die Konferenz einzuladen würde nicht gehen, weil dann Wissenschaftler B und C Popularität 2 hätten, und die Bedingung oben wäre für B und C dann $1 + 2 < \frac{6}{2}$, was falsch ist. Wir könnten aber Wissenschaftler A, C, D, E und F einladen. Die Popularität der Wissenschaftler C, D, E und F wäre dann 1, die Popularität von Wissenschaftler a wäre 0, und sowohl $1 + 1 < \frac{5}{2}$ als auch $1 + 0 < \frac{5}{2}$ sind wahr.

Maus Binna ist es egal wie viele Wissenschaftler an die Konferenz eingeladen werden, solange es mindestens einer ist: $k > 0$. Kannst du ihr helfen und herausfinden, welche Wissenschaftler sie einladen soll, oder feststellen, dass es unmöglich ist, so eine Konferenz zu organisieren? Falls es mehrere Mengen von Wissenschaftlern gibt, welche die obige Bedingung erfüllen, kannst du eine beliebige dieser Mengen auswählen.

Aus ihrer bisherigen Erfahrung im Organisieren von Konferenzen weiss Maus Binna, dass es einfacher ist, so eine Menge von Wissenschaftlern zu finden, wenn die Anzahl Paare von Wissenschaftlern, die gemeinsam einen Artikel veröffentlicht haben, klein ist. Deshalb werden bei dieser Aufgabe Lösungen belohnt, die nur in Fällen funktionieren wo m relativ klein ist, und Lösungen, die eine bessere Laufzeit haben, wenn m relativ klein ist. Siehe die Teilaufgaben weiter unten für mehr Details.

In jeder der Teilaufgaben 2-4 musst du deinen Algorithmus beschreiben, den *Pseudocode*, die *Laufzeit* und *Speichernutzung* aufschreiben, sowie dessen *Korrektheit* erklären.

Um für jede der Teilaufgaben 2-4 volle Punktzahl zu erhalten, muss deine Lösung korrekt sein und gute Laufzeit und Speichernutzung haben für die Fälle wo m die Einschränkungen

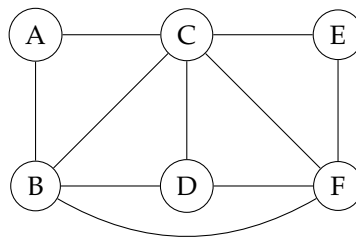
der Teilaufgabe erfüllt. Es gibt aber bereits für unfertige oder suboptimale Lösungen in jeder Teilaufgabe Punkte.

Du darfst Lösungen schreiben, die alle oder mehrere Teilaufgaben gleichzeitig lösen, oder für verschiedene Teilaufgaben verschiedene Lösungen schreiben.

In allen Teilaufgaben kannst du annehmen, dass $n \geq 5$.

Teilaufgabe 1: Löse ein Beispiel (10 Punkte)

Es ist folgende Menge von Wissenschaftlern gegeben, wobei die Kreise Wissenschaftler repräsentieren, und Wissenschaftler, die gemeinsam einen Artikel veröffentlicht haben, durch Linien verbunden sind. Kannst du eine Menge von Wissenschaftlern finden, die zu einer Konferenz eingeladen werden können?



Um Missverständnisse zu vermeiden, ist hier die gleiche Menge von Paaren von Wissenschaftlern, die gemeinsam einen Artikel veröffentlicht haben, in Textform aufgeschrieben:

$$\{\{A, B\}, \{A, C\}, \{B, C\}, \{B, D\}, \{C, D\}, \{C, E\}, \{C, F\}, \{D, F\}, \{B, F\}, \{E, F\}\}$$

Du musst deine Antwort nicht beweisen, es genügt, eine beliebige korrekte Menge aufzuschreiben.

Teilaufgabe 2: Sehr wenige Artikel (20 Punkte)

Die Anzahl Paare von Maus-Wissenschaftlern, die gemeinsam einen Artikel veröffentlicht haben, ist sehr klein: $m < \frac{(n-5)^2}{10}$.

Teilaufgabe 3: Wenige Artikel (30 Punkte)

Die Anzahl Paare von Maus-Wissenschaftlern, die gemeinsam einen Artikel veröffentlicht haben, ist relativ klein: $\frac{(n-5)^2}{10} \leq m < \frac{(n-5)^2}{4}$.

Teilaufgabe 4: Viele Artikel (40 Punkte)

Die Anzahl Paare von Maus-Wissenschaftlern, die gemeinsam einen Artikel veröffentlicht haben, ist relativ gross: $m \geq \frac{(n-5)^2}{4}$.

Einweg-Portale

Mouse Stofl wurde kürzlich zum Transportminister in der Allianz der unabhängigen Systeme (kurz *die Allianz*) gewählt.

Die Allianz besteht aus n Planeten, wobei Planet Apodemus die Hauptstadt ist. Jeder Planet war ursprünglich von je genau einer Maus bewohnt. Unter dem Projekt "Alle Wege führen nach Apodemus" versuchte Stofls Vorgänger, alle Mäuse dazu zu bewegen, in die Hauptstadt zu ziehen. Er baute dazu $n - 1$ Einweg-Portale, so dass man von jedem Planeten aus Apodemus erreichen kann, direkt oder via andere Planeten. Die Mission schlug jedoch fehl: Gewisse Planeten sind nun zwar unbewohnt, andere jedoch sind immer noch von je einer einzelnen Maus bewohnt, welche nicht nach Apodemus ziehen wollte. Dies führte auch dazu, dass Stofls Vorgänger nicht mehr wiedergewählt wurde.

Um die verbliebenen Mäuse davon zu überzeugen, nach Apodemus zu ziehen, überlegt sich Maus Stofl, neue Portale zu bauen. Da diese nicht billig sind, möchte er maximal k davon bauen. Das Ziel dieser Aktion ist, die durchschnittliche Anzahl Portale zu minimieren, die man benötigt, um nach Apodemus zu kommen, wobei man nur die noch bewohnten Planeten einberechnet. Stofl weiss, welche Portale welche Planeten verbinden, und welche Planeten noch bewohnt sind, weiss allerdings nicht, was die beste Möglichkeit ist, die k neuen Portale zuzufügen. Kannst du ihm helfen?

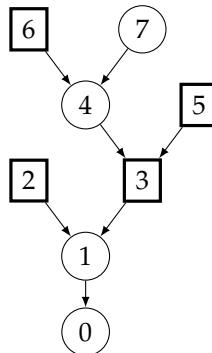
Formeller gesagt, kann man die Allianz als einen gewurzelten, gerichteten und ungewichteten Baum mit n Knoten darstellen, wobei alle Kanten zur Wurzel gerichtet sind. Die Wurzel ist dabei der Planet Apodemus. Eine Teilmenge S der Knoten im Baum nennen wir "von einer einzelnen Maus bewohnt" (wobei Apodemus nicht in S ist). Nun kannst du bis zu k zusätzliche gerichtete ungewichtete Kanten zum Baum zufügen. Sei $f(v)$ die Länge eines kürzesten Pfades von v zur Wurzel, nach dem Zufügen dieser zusätzlichen Kanten, und sei

$$A = \frac{\sum_{v \in S} f(v)}{|S|}$$

die durchschnittliche Anzahl Portale für Knoten in S wenn man diese Kanten benutzt. Finde den kleinsten erreichbaren Wert für A .

Teilaufgabe 1: Löse das Beispiel (10 Punkte)

Stofl möchte, dass du ihm zuerst ein Beispiel vorlöst, damit er weiss, dass du ihm überhaupt helfen kannst. In diesem Beispiel gilt $n = 8$, $k = 2$ und die bewohnten Planeten sind $S = \{2, 3, 5, 6\}$. Apodemus ist der Planet mit der Nummer 0.



Was ist der kleinste erreichbare Wert für A und wo müssen die Portale plziert werden, um diesen Wert zu erreichen? Du musst deine Lösung nicht begründen.

Teilaufgabe 2: Teure Portale, Wenig Geld (20 Punkte)

Die Portale sind sehr teuer und die Allianz kann sich nur ein neues Portal leisten. In dieser Teilaufgabe gilt $k = 1$.

Beschreibe einen Algorithmus, der diese Aufgabe löst, und argumentiere über dessen Korrektheit. Dein Algorithmus soll A berechnen. Du kannst annehmen, dass die Eingabe in einer passenden Form vorliegt (z. B. der Baum als Adjazenzliste, wobei Apodemus dem Knoten 0 entspricht).

Analysiere die Laufzeit und die Speichernutzung abhängig von n .

Teilaufgabe 3: Die Allianz (70 Punkte)

In dieser Teilaufgabe, hilf Stofl die allgemeine Aufgabe zu lösen.

Beschreibe einen Algorithmus, der diese Aufgabe löst, und argumentiere über dessen Korrektheit. Dein Algorithmus soll A berechnen. Du kannst annehmen, dass die Eingabe in einer passenden Form vorliegt.

Analysiere die Laufzeit und die Speichernutzung abhängig von n und k . Versuche die Laufzeit vor der Speichernutzung zu optimieren.

Beachte, dass $k \leq n$ gilt. Dies bedeutet, dass ein Algorithmus mit einer Laufzeit in $O(n^7k)$ schneller als ein Algorithmus mit einer Laufzeit in $O(n^8)$ gilt, aber ein Algorithmus mit einer Laufzeit in $O(nk^2)$ langsamer als ein Algorithmus mit einer Laufzeit in $O(n^2)$ gilt (z. B. in einem Fall mit $n = k$).

Bemerkung: Falls du keinen Algorithmus findest, wird bedeutender Fortschritt in Richtung einer Lösung mit Teilpunkten belohnt.

Teilchendetektor

Maus Binna möchte ein Teilchenphysik-Experiment am Large Hadron Collider durchführen, welcher tief unter der Erde um Genf ist. Zu diesem Zweck hat sie einen Teilchendetektor gebaut, der aus einem Ring von Sensoren besteht. Es gibt N Sensoren mit den Nummern $0, 1, \dots, N - 1$. Die Sensoren 0 und 1 liegen nebeneinander, ebenso wie 1 und $2, \dots, N - 2$ und $N - 1, N - 1$ und 0 .

Leider haben einige Sensoren Herstellungsfehler, welche nicht leicht zu erkennen sind. Deshalb weiss Binna nicht, welche Sensoren kaputt sind; Es könnte sogar sein, dass alle oder keine kaputt sind. Jedoch besitzen die Sensoren einen speziellen Modus, der es ermöglicht, zu erkennen, ob ein benachbarter Sensor defekt ist. Wenn ein benachbarter Sensor defekt ist, meldet er eine 1 , andernfalls eine 0 . Wenn jedoch beide benachbarten Sensoren defekt sind, heben sie sich gegenseitig auf und der Fehler kann nicht erkannt werden. Ist der Sensor, mit dem der Defekt festgestellt wird, selbst defekt, wird das Ergebnis invertiert.

Im Wesentlichen bedeutet dies, dass ein guter Sensor 0 meldet, wenn beide Nachbarn die gleiche Fehlerhaftigkeit aufweisen (entweder sind beide gut oder beide defekt), und ansonsten 1 . Ebenso meldet ein defekter Sensor 1 , wenn beide Nachbarn gleich defekt sind, und sonst 0 .

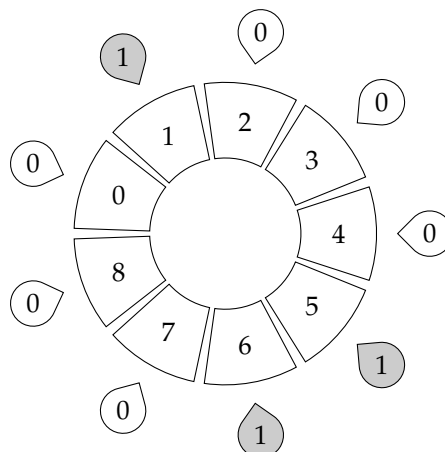
Maus Binna hat die Berichte von allen Sensoren gesammelt. Sie möchte nun wissen, wie viele defekte Sensoren es höchstens geben kann. Es könnte auch sein, dass es gar keine Konfiguration von defekten und guten Sensoren gibt, die zu den gemeldeten Werten führen würde; dies deutet darauf hin, dass mit dem Partikeldetektor etwas anderes nicht stimmt.

Formale Beschreibung Gegeben ist die Anzahl der Sensoren $N \geq 3$, sowie der Bericht $r_i \in \{0, 1\}$ für jeden Sensor $i \in \{0, \dots, N - 1\}$. Gib die höchstmögliche Anzahl von defekten Sensoren aus, oder -1 , wenn es keine Konfiguration von Sensoren gibt, die zu den gegebenen Berichten führen würde.

Teilaufgabe 1: Miniatur Detektor (15 points)

Maus Binna hat einen kleinen Detektor gebaut, der wie folgt aussieht. Die umkreisten Zahlen sind die Meldungen der Sensoren.

Kannst du herausfinden, wie viele kaputte Sensoren es höchstens geben kann?





Teilaufgabe 2: Alle Sensoren melden 1 (10 points)

Alle Sensoren des Detektors melden 1.

Berechne die höchstmögliche Anzahl defekter Sensoren für ein beliebiges N , wenn $r_i = 1$ für alle i gilt. Es reicht, einen Ausdruck aufzuschreiben, du musst keinen Algorithmus beschreiben. Zeige warum diese Antwort korrekt ist.

Teilaufgabe 3: Alle Sensoren melden 0 (15 points)

Alle Sensoren des Detektors melden 0. Ist wirklich alles in Ordnung?

Berechne die höchstmögliche Anzahl defekter Sensoren für ein beliebiges N , wenn $r_i = 0$ für alle i gilt. Es reicht, einen Ausdruck aufzuschreiben, du musst keinen Algorithmus beschreiben. Zeige warum diese Antwort korrekt ist.

Teilaufgabe 4: Allgemeinen Fall (60 points)

Jetzt gibt es keine Einschränkungen mehr. Beschreibe einen Algorithmus, welcher für einen beliebigen Bericht die höchstmögliche Anzahl defekter Sensoren berechnet.